数理解析研究所講究録
1038 巻 1998 年 127-134

浮動小数係数多変数多項式の終結式計算における桁落ち誤差

佐藤 智之 (筑波大学大学院 教育研究科)
佐々木 建昭 (筑波大学 数学系)

Abstract. 浮動小数係数多変数多項式の終結式を計算する場合、用いる算法によって大きな桁落ちが起きたり起きなかったりする。よく知られている算法は、大きな桁落ちを起こす、あるいは高次の場合に計算量が著しく大きくなる、のいずれかである。本稿では、1) 大きな桁落ちを起こすのは多項式除算を含む算法であり、2) 佐々木・村尾の “効率的ガウス消去法” を用いる算法は大きな桁落ちを起こさず、漸近的な計算量もほどほどであることを実例により示す。

1. はじめに

1996 年の数式処理学会において、照井・佐々木が「浮動小数係数 2 変数多項式の終結式計算において、部分終結式算法は著しい桁落ちを起こすが、Bezout 行列式を小行列式展開する算法はほとんど桁落ちを起こさない」ことを指摘した [T&S96]。しからば、終結式はすべて Bezout 行列式を小行列式展開して計算すればよさそうだが、小行列式展開法は与式が高次のとき著しく時間がかかる。桁落ちが少なく、漸近的計算量がほどほどになる算法はないのだろうか。本稿は部分的にではあるが、この疑問に答えるものである。

3 章での実験例が示すように、終結式計算の場合、桁落ちは専ら多項式除算で生じる。したがって、多項式除算を用いないで、あるいは桁落ちが起きにくい形で用いて、計算ができればよいと考えられる。そのような行列式算法として、佐々木・村尾が 1981 年に発表した効率的ガウス消去法がある。本稿では部分終結式算法と Bezout 行列式に対する各種の行列式計算法を比較し、効率的ガウス消去法が起こす桁落ちが小さいことを実例で示す。

2 章では、多変数多項式に対する代表的な終結式算法を概観する。特に効率的ガウス消去法について説明する。3 章では、a) 終結式計算における桁落ちは専ら多項式除算でひき起こされること、b) 効率的ガウス消去法における桁落ちが少ないこと、を実験例で示す。4 章では実験結果を考察する。

2. 終結式算法の概観

多変数多項式の終結式算法の主なものは以下である。

1) Sylvester 行列式を計算する。

2) Bezout 行列式を計算する [K&A69]。

3) 部分終結式算法で多項式剰余列を計算する [Bro78]。

4) 頭項消去法で主係数を消去する [Sas91]。

5) 1 変数多項式に写像し、1 変数多項式の終結式から補間する [Col71]。

1) と 2) はどちらも行列式を計算するのだが、Bezout 行列式の方が Sylvester 行列式より低次なので、算法 1) は本稿ではとりあげない。3) と 4) はどちらも消去法に基ずくが、3) が主項消去で 4) が頭項消去の違いがある。どちらも一長一短があるが、著者らはこれまで 3) のみ調べてきたので、本稿では 4) は割愛する。5) は整数、有理数係数の多項式に対しては効率のよい方法であるが、浮動小数係数の場合にはどうなるか分からない。興味深い研究対象であるが、現時点までには研究時間が足りず、今後の課題である。

次に、行列式計算に対しては、有名な算法として小行列式展開法 ([Sas81] 参照) とガウス消去法 (Bareiss の算法 [Bar68]) がある。さらに、ほとんど知られていないが ([Sas81] には紹介されている)、効率的ガウス消去法 [S&M82] もある。

本章では、部分終結式算法、Bareiss の算法、および効率的ガウス消去法について簡単に説明する。

2.1. 部分終結式算法

F と G を多項式環 $\mathbf{R} = \mathbf{K}[x, y, \cdots, z]$ の要素とし、

$$\begin{aligned} F &= f_l x^l + f_{l-1} x^{l-1} + \cdots + f_0, \quad f_l \neq 0 \\ G &= g_m x^m + g_{m-1} x^{m-1} + \cdots + g_0, \quad g_m \neq 0 \end{aligned} \tag{1}$$

と表す。F と G の変数 x に関する**終結式**を $\mathrm{res}(F, G)$ と表す。終結式とは F と G から変数 x を消去したものであり、具体的には F と G の多項式剰余列を計算すれば、得られる。多変数多項式に対する剰余列計算法として最良のものは部分終結式算法であるが、その算法で得られる最後の要素が終結式になる。部分終結式算法は以下である [Bro78]。

Alogorithm 1 (subresultant)

Input: 原始多項式 F_1 と F_2, $\deg(F_1) \geq \deg(F_2)$;

Output: $D = \gcd(F_1, F_2)$;

$\delta_1 \leftarrow \deg(F_1) - \deg(F_2)$;
$F_3 \leftarrow \mathrm{prem}(F_1, F_2)/(-1)^{\delta_1+1}$;
$i \leftarrow 3$; $\zeta \leftarrow 1$;
while $\deg(F_i) > 0$ do begin
  $i \leftarrow i + 1$;
  $\delta_{i-2} \leftarrow \deg(F_{i-2}) - \deg(F_{i-1})$;
  $\zeta \leftarrow \{\mathrm{lc}(F_{i-2})\}^{\delta_{i-3}} \zeta^{1-\delta_{i-3}}$;
  $F_i \leftarrow -\mathrm{prem}(F_{i-2}, F_{i-1})/\{\mathrm{lc}(F_{i-2})\zeta^{\delta_{i-2}}\}$ **end**; (2)
return F_i □

(注)上算法中、gcd は最大公約数を、prem は擬剰余を、 lc は主係数をそれぞれ表す。

2.2. Bareiss の算法

M を $n \times n$ 行列とする。$M^{(k)}$ を $(n-k)\times(n-k)$ 行列とし、その ij 要素 $M_{ij}^{(k)}$(ただし、i と j は 1 から $n-k$ まで動くのではなく、$k+1$ から n まで動くとする)を次式とする。

$$M^{(k)} = (M_{ij}^{(k)}), \quad M_{ij}^{(k)} = \begin{vmatrix} a_{11} & \cdots & a_{1k} & a_{1j} \\ \vdots & \ddots & \vdots & \vdots \\ a_{k1} & \cdots & a_{kk} & a_{kj} \\ a_{i1} & \cdots & a_{ik} & a_{ij} \end{vmatrix} \quad (k+1 \le i,j \le n) \tag{3}$$

このとき次の補題が成立する。

Lemma 1 $1 \le k \le n-2$ かつ $k+2 \le i,j \le n$ なる任意の i,j に対し

$$\widetilde{M_{ij}}^{(k+1)} \equiv \begin{vmatrix} M_{k+1,k+1}^{(k)} & M_{k+1,j}^{(k)} \\ M_{i,k+1}^{(k)} & M_{i,j}^{(k)} \end{vmatrix} = M_{kk}^{(k-1)} M_{ij}^{(k+1)} \tag{4}$$

が成立する。(証明は [Sas81] 参照) □

さて、$M = M^{(0)}$ とし、行列 $M^{(0)}$ の第 1 列をガウス消去すると

$$|M| = |M^{(0)}| = \begin{vmatrix} a_{11} & 0 \\ 0 & M^{(1)} \end{vmatrix} / a_{11}^{n-1} = |M^{(1)}|/a_{11}^{n-2}$$

を得る。次に $M^{(1)}$ の第 1 列(上記の i,j の定め方によれば "第 2 列")をガウス消去すると $|\widetilde{M}^{(2)}|/{M_{22}^{(1)}}^{n-3}$ を得るが、(4) 式によれば、$\widetilde{M}^{(2)}$ の各要素は $M_{11}^{(0)} = a_{11}$ で割り切れる。したがって、上式の分母因子 a_{11}^{n-2} は完全にキャンセルされる。同様に、k 回目の消去で生じる分母因子 $(M_{kk}^{(k-1)})^{n-k-1}$ は、$k+1$ 回目の消去後の行列 $\widetilde{M}^{(k+1)}$ の各要素の共通因子と打ち消し合う。したがって、$M_{ij}^{(k+1)} = \widetilde{M}_{ij}^{(k+1)}/M_{kk}^{(k-1)}$ なる算式で $\widetilde{M}^{(k)}$ の代わりに $M^{(k)}$ を計算すれば、分母の因子を考える必要はなくなり、中間式膨張を著しく押えられる。この事実を利用した行列式算法が次に示す Bareiss の算法である [Bar68]。

Alogorithm 2 (Bareiss)

Input: $M(n \times n), \quad n \ge 2;$

Output: $D = |M|;$

for $k = 1$ to $n-1$ do **begin**
  if $k = 1$ **then** $M_{00}^{(0)} \leftarrow 1$ **else** make pivoting so that $M_{k,k}^{(k-1)} \neq 0;$
  for $i \leftarrow k+1$ to n do
    for $j \leftarrow k+1$ to n do

$$M_{ij}^{(k+1)} \leftarrow (M_{k+1,k+1}^{(k)} M_{ij}^{(k)} - M_{i,k+1}^{(k)} M_{k+1,j}^{(k)})/M_{k,k}^{(k-1)}; \tag{5}$$

end;

return $D \leftarrow M_{nn}^{(n)}$ □

2.3. 効率的ガウス消去法

(4) から明らかなように、行列 M の要素が整数あるいは 1 変数多項式の場合には Bareiss 算法の手間はたかが知れているが、多変数多項式の場合の計算は容易ではない。ところで、(5) の除算において必要なのは商のみで、剰余を必要としない除算においては、被除数の低次の項は計算する必要はない。この事実を利用したのが効率的ガウス消去法である。

まず、行列 M の各対角要素が他の要素とは独立な変数である場合を考える。

$$M = \begin{pmatrix} X_1 & a_{12} & \cdots & a_{1n} \\ a_{21} & X_2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & a_{n-1,n} \\ a_{n1} & \cdots & a_{n,n-1} & X_n \end{pmatrix} \tag{6}$$

このとき、行列式 $|M|$ は $X_1, \cdots, X_n$ の各変数に関して 1 次式で、$M_{k-1,k-1}^{(k-2)}$ は次のように書ける。

$$M_{k-1,k-1}^{(k-2)} = X_1 X_2 \cdots X_{k-1} + N^{(k-1)} \tag{7}$$

(ただし N^{k-1} は $X_k, \cdots, X_n$ を含まず、$X_1, \cdots, X_{k-1}$ について $k-2$ 次以下)

ここで、$X_1, \cdots, X_{k-1}$ を主変数とみて $(X_1 \cdots X_{k-1})^2$ を $M_{k-1,k-1}^{(k-2)}$ で割り、商 $Q^{(k-1)}$ と剰余 $R^{(k-1)}$ を計算し、(4) 式と辺々掛け合わせれば次式を得る。

$$M_{ij}^{(k)}(X_1 \cdots X_{k-1})^2 = (M_{kk}^{(k-1)} M_{ij}^{(k-1)} - M_{ik}^{(k-1)} M_{kj}^{(k-1)}) Q^{(k-1)} + M_{ij}^{(k)} R^{(k-1)} \tag{8}$$

$M_{ij}^{(k)}$ は $X_1, \cdots, X_{k-1}$ の各変数に関して 1 次で、$R^{(k-1)}$ には $X_1 X_2 \cdots X_{k-1}$ に比例する項は含まれないので、右辺の第 2 項は $(X_1 \cdots X_{k-1})^2$ に比例する項を含まない。よって (8) の右辺の第 1 項から $(X_1 \cdots X_{k-1})^2$ に比例する項だけを取り出せば、$M_{ij}^{(k)}$ が計算できることが分かる。

効率よく計算するために特殊な乗算 $\otimes$ を導入する。P と $\widetilde{P}$ が主変数 $X_l, X_{l+1}, \cdots$ の多項式で X_l に比例する部分と比例しない部分に分解した時、$P = P_1 X_l + P_0, \widetilde{P} = \widetilde{P_1} X_l + \widetilde{P_0}$ と書けるとする。このとき、演算子 $\otimes$ を次式で定める。

$$P \bigotimes_l^k \widetilde{P} = \begin{cases} (P_1 \bigotimes_{l+1}^k \widetilde{P_1}) X_l + (P_1 \bigotimes_{l+1}^k \widetilde{P_0} + P_0 \bigotimes_{l+1}^k \widetilde{P_1}) & (k \geq l) \\ P \widetilde{P} & (k < l) \end{cases}$$

この乗法を用いれば、$Q^{(k-1)}$ と $M_{ij}^{(k)}$ は次のように計算できる。

$$Q^{(k-1)} = X_1 X_2 \cdots X_{k-1} - N^{(k-1)} + N^{(k-1)} \bigotimes_1^{k-1} N^{(k-1)} - N^{(k-1)} \bigotimes_1^{k-1} N^{(k-1)} \bigotimes_1^{k-1} N^{(k-1)} + \cdots \tag{9}$$

$$M_{ij}^{(k)} = (M_{kk}^{(k-1)} \bigotimes_{1}^{k-1} M_{ij}^{(k-1)} - M_{ik}^{(k-1)} \bigotimes_{1}^{k-1} M_{ij}^{(k-1)}) \bigotimes_{1}^{k-1} Q^{(k-1)} \tag{10}$$

ただし、$N^{(k-1)}$が $X_1, \cdots, X_{k-1}$ のすべての変数につき $k-2$ 次以下の項しか含まないので、(10) においては $\otimes$ を $k-2$ 回作用させた段階で終了する。なお、一般には対角要素が独立変数となることはあり得ないので、計算に先立って対角要素を独立変数で置き換える必要がある。その際、X_1 から X_{n-2} まで置き換えれば十分である。以上により次の算法が得られる [S&M82]。

Alogorithm 3 (Efficient-Gaussian)

Input: $(n \times n)$ matrix M such that variables $X_1, \cdots, X_{n-2}$ are not contained in M;

Output: $D = |M|$;

Step1 [Replace diagonal elements by new variables]
for $k \leftarrow 1$ to $n-2$ replace M_{kk} by X_k;

Step2 [Eliminate columns]
for $k \leftarrow 1$ to $n-1$ do begin
if $k = 1$ **then** $Q^{(k-1)} \leftarrow 1$
else begin

$$N^{(k-1)} \leftarrow M_{k-1,k-1}^{(k-2)} - X_1 X_2 \cdots X_{k-1};$$

$$Q^{(k-1)} \leftarrow X_1 X_2 \cdots X_{(k-1)} - N + N \bigotimes_{1}^{k-1} N - N \bigotimes_{1}^{k-1} N \bigotimes_{1}^{k-1} N + \cdots;$$

end;
for $i \leftarrow k+1$ to n
for $j \leftarrow k+1$ to n

$$M_{ij}^{(k)} \leftarrow (M_{kk}^{(k-1)} \bigotimes_{1}^{k-1} M_{ij}^{(k-1)} - M_{ik}^{(k-1)} \bigotimes_{1}^{k-1} M_{ij}^{(k-1)}) \bigotimes_{1}^{k-1} Q^{(k-1)}$$

end;

Step3 [Recover original polynomial]

for $k \leftarrow 1$ to $n-2$ substitute M_{kk} for X_k in M_{nn};

Step4 **return** $D \leftarrow M_{nn}$ □

3. 実験

我々は、上記の四つのアルゴリズム、すなわち部分終結式算法、Bezout の行列式に対する小行列式展開法、Bareiss の算法、および効率的ガウス消去法を数式処理システム**GAL**に実装し、それぞれの方法における計算結果の数値精度を調べた。具体的には係数を有効浮動小数 (effective-float,[KS97] 参照) で表しておき、多項式の係数全体の最大精度と最小精度をとり出すコマンド "precminmax" を用いて精度を調べた。

消去回数	最大相対誤差
1	4.0e-14
2	0.000000000018765628255216
2'	0.0000000014680064000103
3	0.0016048616984351
3'	0.002089306890789

表 1. 部分終結式算法での相対誤差

(算法)	最小相対誤差	最大相対誤差
部分終結式算法	1.0e-15	0.002089306890789
ガウス消去法	1.04327843738032e-15	0.000000000027893759991193
小行列式展開法	3.0510055981754e-16	4.0e-15
効率的ガウス消去法	4.2452830188679e-16	4.0e-15

表 2. P_1, P_2 におけるそれぞれの算法での最大、最小相対誤差

(算法)	最小相対誤差	最大相対誤差
部分終結式算法	1.0e-15	0.98687189544955
ガウス消去法	1.0e-15	6.9078947368449e-13
小行列式展開法	1.2967966036279e-16	8.8815789473682e-14
効率的ガウス消去法	1.1949054419143e-16	5.9210526315789e-14

表 3. P_3, P_4 におけるそれぞれの算法での最大、最小相対誤差

実験には次の方程式を用いた。(ただし、係数は有効浮動小数に変換される)

$$
\begin{aligned}
P_1 &= ((y+1)x^4 + (2y^2-3)x^2 + (y^2+y-3)x - (y^3-2y^2+y))/3 \\
P_2 &= ((y^2+2y-1)x^3 + (y-1)^2x + (y^3-3y^2+y+5))/3 \\
P_3 &= ((z-2)(y^2+1)x^3 + (z^2-5)(2y-3)x^2 + (z-1)(y^2+y-3)x - (z-5)(2y^2+1))/7 \\
P_4 &= ((y^2+2y-1)x^3 + (y-1)^2x + (y^3-3y^2+y+5))/11
\end{aligned}
\tag{11}
$$

実験 1 部分終結式算法での剰余列計算における相対誤差
P_1 と P_2 を出発多項式として、Algorithm1 で剰余列を計算するときの係数部の最大相対誤差を示したのが表 1 である。ここで、消去回数 2 と 3 には擬剰余 $\mathrm{prem}(P_2, P_3)$ と $\mathrm{prem}(P_3, P_4)$ の結果を示し、消去回数 2' と 3' には prem に含まれる共通因子を除く徐算での結果を示した。剰余を計算するたびに非常に大きな桁落ちが生じることが分かる。

実験 2 $\mathrm{res}(P_1, P_2)$ 計算における桁落ち誤差の比較
表 2 は、$\mathrm{res}(P_1, P_2)$ 計算を部分終結式算法、Bezout 行列式をガウス消去法、小行列式展開法、効率的ガウス消去法で計算した時の最大相対誤差、最小相対誤差を示したものである。これによると、部分終結式算法に比べてガウス消去法は桁落ち誤差を大きく低減させるが、それでも誤差はかなり大きい。それに対して小行列式展開法と効率的ガウス消去法における桁落ち誤差は無視できるほど小さいことが分かる。

実験 3 $\mathrm{res}(P_3, P_4)$ 計算における桁落ち誤差の比較
表 3 は、3 変数多項式に対する四つの算法の桁落ち誤差を比較したものである。ここで、部分終結式算法の桁落ち誤差が異常に大きくなっているのは、おそらく正確にキャンセルすべき項が浮動小数計算で微小項として残ったためである。3 変数多項式の場合も、小行列式展解法と効率的ガウス消去法ではほとんど桁落ちが起きないことが分かる。

4. 実験結果の考察と今後の課題

何故、部分終結式算法に比べて、小行列式展開法や効率的ガウス消去法が桁落ちが少ないのであろうか。剰余の計算は主項消去の繰り返しである。ここで、主項消去とは次式で定義される。

$$F \text{ の } G \text{ による主項消去} = F - \mathrm{lc}(F)/\mathrm{lc}(G)\, x^{\deg(F)-\deg(G)} \times G$$

この式によると、$r = |\mathrm{lc}(F)/\mathrm{lc}(G)| \gg 1$ のとき、G の誤差部が r 倍に拡大され、その分、F の方の精度が相対的に失われることが分る。すなわち、剰余計算では大きな桁落ちが起きやすいのである。これに対し、正確に割り切れるときの商の計算は、それほど大きな桁落ちを起こさないことが表 1 の 2' と 3'、および表 2 のガウス消去法のデータより分かる。それでも、商の計算においてもある程度の桁落ちが生じることは避けられない。そのことはガウス消去法に対するデータからも見てとれる。しかしながら、効率的ガウス消去法では、除多項式は (7) の形で主係数が常に 1 である。したがって、一般の商計算に比べてはるかに桁落ちが生じにくいのである。

今後の最大の課題は算法の誤差解析である。たとえば、小行列式展開法においてさえ、項が互いに打ち消し合うことがあるが、浮動小数係数では正確な打ち消しと偶然的に係数が小さくなる場合とを区別できない。計算においてどれほど小さな項が現れるか、微小項の効果は、など誤差解析の課題は多い。

なお、上記算法に対する現在のプログラムは第 0 バージョンで、タイミングデータを云々できるような代物ではない。今後は種々の効率化のアイディアを組み込み、プログラムを完成させていく予定である。

参 考 文 献

[Bar68] Bareiss, E. H., Sylvester's identity and multistep integer-preserving Gaussian elimination. Math. Comput., 22 (1968), 565–578.

[Bro78] Brown, W. S., The subresultant PRS algorithm, ACM Trans. Math Soft., 4 (1978), 237–249.

[Col71] Collins, J. E., The calculation of multivariate resultant, J. ACM, 18 (1971), 515–532.

[K&A69] Ku, S. Y. and Adler, R. J., Computing polynomial resultants: Bezout's Determinant vs. Collins' reduced PRS algorithm, C. ACM, 12 (1969), 23–30.

[KS97] Kako, F. and Sasaki, T., Proposal of "effective" floating-point number, preprint (May 1997), 12 pages (submitted for publication).

[Sas81] 佐々木 建昭, 数式処理, 情報処理叢書. 情報処理学会, 1981.

[Sas91] 現代数理科学辞典 (編集代表・広中平祐), XVIII 基礎数理 (I), [4] 計算機代数 (佐々木建昭執筆), 大阪書籍, 1991

[S&M82] Sasaki, T. and Murao, H., Efficient Gaussian elimination method for symbolic determinants and linear systems, ACM Trans. Math. Soft., 8 (1982), 277–289.

[Tak65] 高木貞治, 代数学講義 (改訂新版), 共立出版, 1965.

[T&S96] 照井章, 佐々木建昭, 浮動小数係数多項式の終結式計算について, 数式処理, 5, 1 (1996), 50–51.

[vdW37] van der Waerden, B.L., Moderne Algebra I, 2nd ed., 1937. (翻訳) 銀林 浩, 現代代数学 1, 東京図書, 1974.